Requirements Analysis Document

Introduction

Purpose

User Stories

### Sprint 1

|  |  |  |
| --- | --- | --- |
| Identifier | User Story | Size |
| ET-1 | As an EGO member I must be able to upload two CSV files to view courses. | 20 |
| ET-2 | As an EGO member I must be able to select courses that I want to be scheduled | 10 |
| ET-3 | As an Ego member I must be able to generate and View a timetable for **Courses** in a way that minimises the number of clashes | 40 |
| ET-4 | As an EGO Member I must be able to set a limit on the number of sessions I want scheduled to restrict the length of the exam period | 10 |
| ET-6 | As an EGO Member I must be able to set a parameter that defines a clash in order to reduce the number of sessions | 10 |
| ET-7 | As an EGO Member I must be able to select a sorting scheme that allows me to prioritise different paramters of the timetable | 10 |

Sprint 2

**User Stories**

|  |  |  |
| --- | --- | --- |
| Identifier | User Story | Size |
| ET-1 | As an Ego member I must be able to generate and View a timetable for ***Exams*** in a way that minimises the number of clashes |  |
| ET-2 | As an EGO member I should be able to enter a list of courses and find those courses position in the timetable |  |
| ET-3 | As an EGO member I must be able to get the timetable for a particular student in Calendar Format to see should the student enquire |  |
| ET-4 | As an EGO member I must be able to see the students with the worst timetables so that I can make provisions for deffered or Supplementary Exams |  |
| ET-5 | As an EGO member I should be able to view courses that interact with a particular course so that I can interact with the relevant course co-ordinators to make arrangements |  |

**Functional Requirements:**

***Upload CSV Files*:**

The system should let a user upload two mandatory CSV files.

On first use the system can not function without both files.The first CSV file is a list of all courses in the University. The second CSV file is one that has a list of all the registered students and the courses that they are enrolled in.Upon uploading the CSV files the system does a check to see if there are courses that have no students enrolled or if a student is enrolled in a course that doesn’t exist. This data is then stored in a secure online database.

***Select courses to be scheduled*:**

The system’s interface should provide a way of selecting courses that the user needs to schedule.

***Generate a timetable:***

The system should have a button that when clicked generates a timetable of exams and displays it for the user to see. This timetable should show the session and all the exams in that session. This timetable has the minimum number of sessions possible without there be any clashes.

**Set the clash parameter:**

The clash parameter is the number or proportion of students that two courses must share to constitute a clash therefore not allowing those courses to occupy to be written at the same time.

The system should provide an input box that allows the user to enter either an integer or an integer percentage ie. 10 or 10%.

This absolute parameter means that if there are 10 or more students shared between two courses then there is a clash.

The relative clash parameter implies that if 10% or more of the students in both courses share the course then that constitutes a clash.

If the clash parameter is not defined it is set 1

**Set a limit on the number of exam sessions:**

The system must provide an input box that accepts only integers greater than zero. The value of this input box is the maximum number of sessions that the timetable can have. The system must then increase the clash parameter until the number of sessions in the generated timetable is less than or equal to the max sessions set.

**Enter a list of courses and find those course’s position in the timetable:**

The system must allow the user to enter or select a subset of courses and find in which sessions those courses appear. This is equivalent to adding a hypothetical student and determining their timetable.

**Students with the worst timetables:**

Upon having generated a timetable the system must be able to return a list of the top n students with the worst timetables(with the most amount of back to back exams)

**View Course interactions:**

The system should when prompted be able to view all courses that interact with a paticular courses(courses it shares students with) in calendar format.

**Non-functional requirements:**

Optimality

Security

Performance

Reliability